Towards An Efficient Key Management and Authentication Strategy for Combined Fog-to-Cloud Continuum Systems

Sarang Kahvazadeh*, Xavi Masip-Bruin*, Rodrigo Diaz‡, Eva Marín-Tordera*
Alejandro Jurnet*, Jordi Garcia*

*Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), Spain
{skahvaza, xmasip, eva, ajurnet, jordig}@ac.upc.edu

‡ Cybersecurity Lab, Atos Spain
rodrigo.diaz@atos.net

***Abstract*— Fog-to-cloud systems have emerged as a novel concept intended to improve service performance by considering fog and cloud resources in a coordinated way. In such a heterogeneous scenario, security provisioning becomes necessary, hence novel security solutions must be designed to handle the highly distributed fog-to-cloud nature. In the security area, key distribution and authentication are referred to as two critical pillars for a successful security deployment. Unfortunately, traditional centralized key distribution and authentication approaches do not meet the particularities brought by a Fog-to-cloud system due to its distributed nature. In this paper, we propose a novel distributed key management and authentication (DKMA) strategy to make Fog-to-cloud systems as secure as possible. The paper ends up presenting some results assessing the benefits of the proposed strategy in terms of traffic and delay reduction.**

Keywords*— *IoT, cloud computing, fog computing, fog-to-cloud computing, security, key distribution and authentication

I. INTRODUCTION AND MOTIVATION

Nowadays, the deployment of edge devices, from sensors and actuators to smart phones and laptops, is increasing day by day worldwide. When these devices embed connectivity and some sort of smart capacities, the whole scenario is referred to as the Internet of Things (IoT). A key benefit brought by IoT is its capacity to develop novel services leveraging wide communication as well as high data collection and processing capacities. The envisioned requirement in high data collection capacity enforces IoT to rely on cloud computing to guarantee a proper data processing –also storage if needed. Cloud computing [1] facilitates an on-demand access to a shared pool of resources, to enable high processing or large storage capacities. However, cloud computing may not be the proper paradigm for services requiring highly constrained demands in terms of, for example, latency for dependable e-health services or immediate decision making processes in industry. Thus, IoT services may look for proximate resources to support services requiring such a specific constraint. To that end, fog computing [2] was proposed to provide real-time processing, low-latency, storage and decision making close to the users demanding the service. Undoubtedly, fog does not come to replace cloud computing, rather fog and cloud must cooperate and thus work under a coordinated umbrella to improve IoT services performance. Aligned to this coordinated scenario, two research trends have recently come up, Fog-to-cloud (F2C) [3] and the OpenFog Reference Architecture (OFRA) [4], both proposing a solution intended to coordinate the resources continuum, from fog to cloud, in a coordinated way. Be it F2C or OFRA, one of the main concerns in this coordinated scenario is the security. This paper deals with security issues in the highly attractive scenario built by considering a joint fog/cloud resources system, such as F2C or OFRA. Hereafter, and for the sake of scenario illustration, we will focus on F2C to represent such a combined resources system.

Related to the security area, key distribution and authentication are preliminary approaches to provide secure communication and integrity in a system. However, traditional cloud keys distribution and authentication, or even the use of one public key generator (PKG), are not foreseen as proper approaches to be applied to a F2C system, mixing up the traditional cloud along with a set of heterogeneous and dynamic fog resources. Indeed, although cloud computing is per se a distributed system (different instances can be allocated to run a service in a transparent, scalable, open and reliable way), in this paper we consider cloud as a centralized approach when compared to fog computing. Thus, hereon we will refer to centralized and distributed approaches when considering traditional cloud or F2C approaches respectively.

Although security is a common and mandatory requirement in general IoT systems, some scenarios relying on IoT devices are extremely sensitive to security concerns. Particularly, we want to explicitly highlight the impact security provisioning may have on a critical infrastructure (CI) system, nowadays enriched with several IoT devices, that while bringing new capacities and features, they also make the system more vulnerable. Nowadays, CI systems are ever relying on many different and heterogeneous IoT devices intended to sensor, detect, monitor and also immediately actuate on the infrastructure (see [5], [6] and [7] for detailed information in this domain). Thus, the specific security weaknesses inherent to IoT devices integrated into CI systems drive several open challenges –key management, key distribution and authentication delay time, network overhead, latency and scalability– that must be carefully addressed.

Some contributions have been already proposed for key distribution and authentication in fog computing, later

reviewed in this paper to learn from the existing literature, although these approaches do not consider the coordinated cloud and fog scenario. One of the first attempts to address the key distribution and authentication in a coordinated F2C scenario was proposed in [8], where a new security architecture leveraging a centralized controller in cloud and distributed controllers in different areas has been proposed. In this paper, we propose a distributed key management and authentication (DKMA) workflow to be applied to the architecture in [8], aiming to illustrate that the security architecture proposed in [8] (using distributed controllers) for key distribution and authentication is much more efficient than a centralized one (cloud) in terms of: i) traffic to the cloud; ii) time spent by the key distribution and authentication processing; iii) network delay and, finally; iv) network overhead. For comparison purposes and in order to illustrate the differences in efficiency assuming the same security quality, we keep the same key distribution and authentication strategy, namely elliptic curve digital signature algorithm, for both scenarios (cloud centralized, and the proposed distributed controllers).

The paper is structured as follows. Section 2 describes the related work; section 3 describes the whole scenario, including the security architecture, the concept of elliptic curve signature cryptography and several tentative approaches for allocating the set of distributed controllers. Then, section 4 presents the proposed DKAM workflow that is evaluated in section 5. Finally, section 6 concludes the paper.

II. Related Work

Several works have been proposed for key distribution and authentication in the cloud arena, hence in a centralized way and thus with no direct applicability to the distributed F2C scenario. However, we revisit such contributions to learn on past efforts.

Authors in [9] propose an identity authentication-based data access control, where key distribution, mutual authentication, and access control are all managed by an authorized agency. This proposal relies on a centralized authorized agency, thus if the authorized agency is compromised, the whole system would be compromised as well (i.e., single point of failure). Similarly, in [10], authors propose an identity-based signcryption scheme with efficient revocation access control for big data, leveraging a centralized PKG for key distribution between users and the analytical system. In [11], authors propose an identity based signcryption scheme with proxy re-encryption access control, also relying on a centralized PKG between users and cloud. Authors in [12], propose to use a trusted third-party system between users, data owners and cloud storage to distribute keys, handle encryption and decryption, as well as authentication. The proposal uses a centralized trusted authority (TA) between cloud and users; hence any compromise in the centralized TA can affect the whole system. In [13], authors propose to use a centralized cryptographic server (CS) aimed at providing each data-file with symmetric keys. The CS provides integrity, confidentiality, access control, and data sharing to the users. Unfortunately, as a centralized approach, it also becomes a single point of failure.

Despite the fact that all reviewed proposals dealing with cloud security can certainly use cloud replication (see for example [14], [15], 16] and [17]), some security issues remain unsolved, such as the latency of the system when using cloud, the complexity added by handling key management for thousands of devices, the huge number of messages forwarded to the cloud, or the increased network delay.

On the other hand, there are some proposals dealing with key distribution and authentication specially oriented to fog computing. These proposals suggest the use of either a centralized public key generator (PKG) or distributed servers to generate keys (usually these servers carry out many different tasks and cannot be so effective to be used as key generators). A review of security proposals in the fog related arena is provided next.

The work in [18] proposes mutual authentication for edge devices, fog and cloud, leveraging a registration authority (RA) located at cloud and already authenticated to the fog servers. Then, the fog servers chose a unique ID (identity) signed with the RA signature and sent to the users. In parallel, when fog users register to the cloud, the RA sends a long-lived random master secret key to the fog users, so that fog users and fog servers can authenticate themselves. The main weakness of this proposal is twofold. First, it relies on a RA centralized at cloud, thus, if the centralized RA is compromised, the whole system can be compromised as well. Second, it uses long-term keys that would not be updated or revoked. Authors in [19] propose a key exchange protocol based on policy-attribute encryption to provide confidentiality, authentication and access control to fog servers. In their architecture, a centralized key generator server is responsible for distributing keys to all entities, while cloud is responsible for defining access control policies. Again, the whole system can be compromised if the centralized key generator server is compromised. In [20], authors propose first to group users, and then a cloud service provider generates keys for these groups. In this architecture, IoT nodes can act as both, clients and servers to generate keys and to establish mutual authentication and secure communication between different user groups. In this proposal, the fact that an IoT node can act as client, server, and also can generate keys, facilitates a potential attacker to act as a server, hence establishing communication with a node in the group and thus compromising the whole group.

According to both the literature review and the envisioned F2C scenario, we argue that a new strategy for key distribution and authentication among fogs and between cloud and fogs is required in F2C systems, due to its distributed nature. We also argue that our proposal, based on a centralized controller in cloud and a set of distributed controllers in different fog areas, can be more efficient (in terms of messages forwarded to cloud, time required for key distribution and authentication, network delay and network overhead), than existing centralized proposals. Moreover, an additional advantage brought in by our proposal, refers to its distributed nature. Indeed, by using the distributed control approach for key distribution and authentication, an undesired problem on a single controller will not compromise the whole system –the centralized F2C controller can revoke the compromised controller and later inform the other. Finally, it is also worth mentioning that the

level of security can, even, be increased by using hybrid key distributions and authentication mechanisms for the different layers (cloud and fog). In fact, this is posed as the next step in the proposed architecture.

III. SCENARIO DESCRIPTION

A. The security architecture

In [8] we introduced for the first time a decoupled architecture intended to provide security to a F2C system, as briefly depicted in Figure 1. In short, the proposed architecture includes a centralized F2C controller located at cloud and several distributed control-area-units (CAU) located at distinct fog areas (areas definition is out of the scope of this paper, see [21] for details). The CAUs would be registered, authenticated and authorized during the F2C initialization phase from a F2C controller located at cloud, thus guaranteeing security in their corresponding areas. It is worth highlighting the fact that a CAU is only responsible for providing security in the devices located within its area. This security architecture brings some remarkable advantages, such as:

- Decreasing the number of messages going to cloud.
- Enhancing privacy and security.
- Providing secure mobility due to the CAUs intercommunication –any attack or failure in CAUs would not affect the whole system because it would be detected and revoked by the F2C controller.
- Key management scalability
- One centralized key generator for CAUs and distributed key generator for providing device's keys in two levels can facilitate authentication.
- Using distributed controllers to improve efficiency in terms of network overhead and energy.
  Be able to use hybrid and different key distribution and encryption at different levels.
- Reducing the entire delay in the system.
- Facilitating key management by using distributed CAUs, so as each device's public keys are not stored at cloud, but stored in their corresponding CAUs instead.

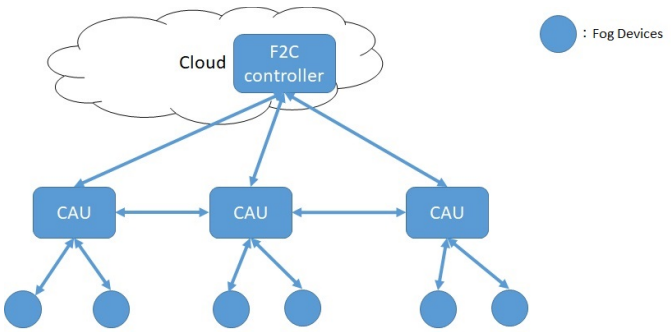

Fig. 1. Proposed security architecture

B. Elliptic Curve Digital Signature

This paper is aimed at comparing the proposed distributed controllers approach vs a traditional cloud centralized approach, in terms of key distribution and authentication delay, and network overhead. To that end, we propose to use elliptic curve key distribution and signature for managing keys and authentication. A key Elliptic Curve Criptography (ECC) advantage is that it provides same security guarantees as Public Key Infrastructure (PKI) and other cryptographies with less key size. In this section we briefly review this concept.

Elliptic curve digital signature algorithm (ECDSA) is known to be an efficient secure certificate-signing algorithm, used in several TLS libraries, such as OpenSSL and GnuTLS. ECDSA depends on modular arithmetic operations on elliptic curves as defined by the equation (1)

$$y^2 \equiv x^3 + ax + b \bmod p \qquad (1)$$

The curve includes three parameters: p, a large prime defining the curve finite field Fp, and the coefficients, a and b [22, 23]. Figure 2 lists all parameters description as used in the algorithm.

Signs	Description
Fp	the curve finite field
p	a large prime defining the curve finite field Fp
a and b	coefficients
G	is the base point of the curve
n	Is the order of G
h	Is the cofactor
sk	Private key
Pk	Public key
k	A uniform random number in the range [1, p-1]
(u,v)	Curve points
r	One part of signature
s	Other part of signature
d	Private key used for signing
z	Hash of the message to be signed
(r,s)	Signature made public

Fig. 2. ECDSA Algorithm sign description

The ECDSA algorithm features the two following functionalities:

1. Key generation: It is based on elliptic curve diffie-hellman, which is used for encryption and decryption.
   1.1. The private key sk is a random integer chosen from $\{1,\ldots,p-1\}$
   1.2. Public key pk is calculated from the curve point multiplication $pk = sk \times G$.
2. Signature and verifying: Used for authentication.
   2.1 Signing:
   - Take a random integer k chosen from range of $\{1,\ldots,p-1\}$
   - A curve point is calculated by: $(u,v) = k \times G$.
   - One part of signature is $r = u \bmod n$.
   - If $r=0$, then choose another k and try again.
   - The other part of signature is $s = k^{-1}(z + rd) \bmod n$.
   - If $s=0$, then choose another k and try again.
   - The pair (r,s) is the signature.

2.2 Verifying: The signer's public key pk, the (truncated) hash z and, obviously, the signature (r,s) are required.

- Calculate the integer $u_1 = s^{-1}z \bmod n$.
- Calculate the integer $u_2 = s^{-1}r \bmod n$.
- Calculate the point $P = u_1G + u_2pk$.

The signature is valid only if $r = u \bmod n$.

C. *Security Architecture Approaches*

In this section, we introduce and analyze different approaches for deploying CAUs along the F2C architecture. Four options are considered, as described next.

- In Figure 3a, we represent the centralized case, where a centralized cloud is responsible for providing security for the whole F2C system. This scheme has some disadvantages, such as: i) the high amount of messages forwarded to cloud; ii) the higher the number of devices the higher the delay; iii) the high network overhead and energy consumption due to the centralized cloud architecture; iv) any compromise or attack in cloud can compromise the whole system.
- A centralized CAU (between cloud and the edge, i.e., closer to users) can be used to provide security for the F2C system (Figure 3b). This approach inherits some of the weaknesses from case 1, such as the centralized view that may put the whole system at risk, when the centralized CAU is attacked.
- Figure 3c implements the scenario where each individual CAU provides security for a single device in the system. This architecture is not desirable mainly for scalability issues.
- Figure 3d extends the scenario presented in Figure 3c by considering each CAU to be attached to several devices at the edge (theoretically within its area). This can be a much more elaborated option to provide security in a F2C system, although certainly bringing several open issues and challenges, such as how many CAUs may control the system, how many devices can be controlled by a CAU, etc.

From the four presented architectures, Figure 3d seems to be the most appropriate one, since different CAUs are responsible for providing security to their corresponding areas. However, extensive evaluations are required to converge on a solid assessment for a single architecture. Moreover, whatever the selected architecture will be, many challenges are certainly demanding further research efforts.

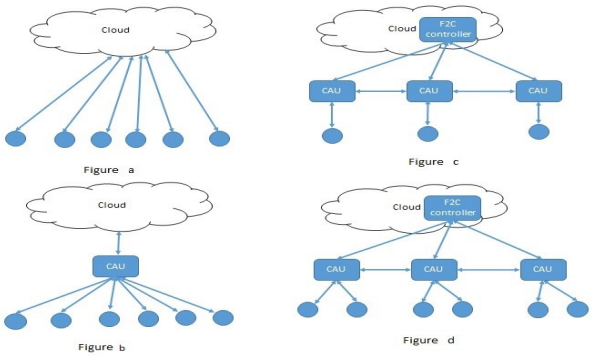

Fig. 3. Different security architecture approaches

IV. THE DKMA PROPOSAL

In this section, we propose a distributed key management and authentication (DKMA) workflow, leveraging the security architecture described in section 3.A. Thus, the main rationale for this paper is twofold. First, to present the DKMA workflow and, second, to show how the deployment of the proposed DKMA in the security architecture briefly summarized in section 3.A increases the efficiency of the key management and authentication processes, in terms of the traffic to cloud, the time demanded by the key distribution and authentication processing, the network delay and the network overhead.

The proposed DKMA assumes CAUs are deployed matching the distributed architecture shown in Figure 3d. The DKMA process assumes one centralized F2C controller and several distributed CAUs. Let's also consider all CAUs will be authenticated and will get keys to provide secure communication to each other but also to the F2C controller. Then all CAUs will also get the authorization from the F2C controller to provide keys and authenticate devices within their areas.

Finally, we assume all devices to be "registered" in a previous registration process, where devices are assigned to a unique "ID". This ID is used to uniquely identify all devices thus preventing a potential attacker to fake itself.

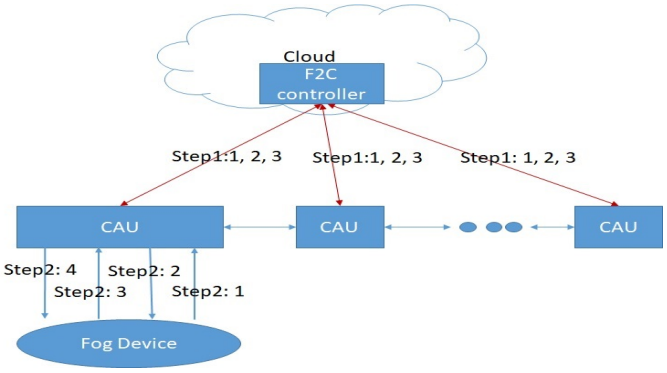

Fig. 4. Distributed key management and authentication

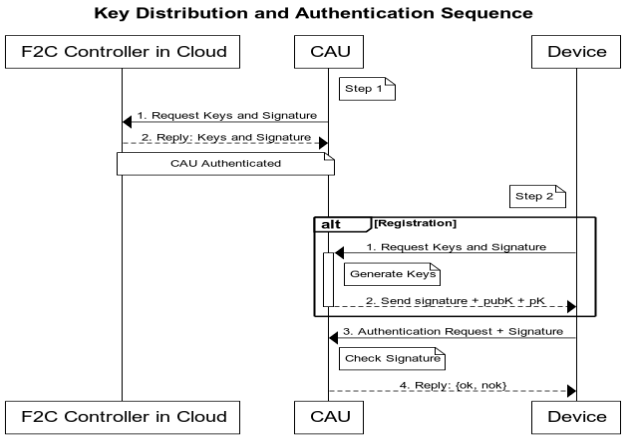

Fig. 5. Distributed key management and authentication workflow

The entire process for the DKMA workflow is shown in Figure 4 and Figure 5 and described next into two main steps:

Step1 (Initialization phase): Control-area unit's (CAU) key distributions and authentication.

1 In the initialization phase of the system, all CAUs get keys and signature from the F2C controller at cloud.

2 The F2C controller authenticates CAUs and provides secure CAUs inter-communication. Therefore, CAUs take the authorization of the F2C controller to generate and provide keys and signature for the devices deployed within their areas.

Step2 (Authentication): All distributed CAUs use the elliptic curve signature based key and signature algorithm to provide keys and authenticate their corresponding devices.

1. Each device requests keys to their corresponding CAUs in their areas.
2. CAUs send keys and signatures to the devices through a secure channel.
3. When a device wants to join the F2C system, it sends the signature to the corresponding CAUs.
4. Then, the corresponding CAU checks, verifies and sends the ACK to the device.

Actions 1 and 2 in step 2 are been done only the first time a device appears in the area. Indeed, if a device leaves the area and come back later to that area, the authentication process only runs actions 3 and 4 in step 2.

In the next Section, we describe the test-bed used for validation purposes, as well as the results obtained to validate the benefits of bringing together the DKMA workflow proposed in this paper in the security architecture previously proposed.

V. Results Analysis

A. Testbed Description

The in-lab scenario proposed to validate the DKMA workflow emulates a smart city with different devices (see Figure 6), such as cars, traffic lights, mobile phones, and so on, by deploying a real test-bed with real (Raspberry Pi) and virtual devices (Virtual Machines). The current test-bed deployment leverages an access point providing connectivity to the environment through the same network. Traffic to the cloud is sent through the router along the link to cloud. A frontend also deploys to manage the test-bed settings as well as to show an overview of the different trials running in the test-bed.

Regarding the network analysis, the test-bed also includes some scripts for packets tracking, thus getting updated information about the network state using a packet catcher (e.g., tcpdump for Linux scenarios) and application logs.

The experimental environment is deployed the proposed DKMA workflow for both the distributed and centralized architectures in a Fujitsu Primergy TX300 S8, hosting 100 virtual devices, and considers a single PC to deploy the centralized cloud approach.

Fig. 6. Test-bed scenario

In the centralized key distribution and authentication approach, an additional computer acts as a cloud responsible for generating keys and signature, distributing them to 100 virtual devices and finally authenticating the devices (sequence described in Figure 4). In the distributed approach instead, 5 computers are distributed as CAUs, and an additional computer acts as cloud. All CAUs are authenticated in the initialization phase, getting the authorization from cloud. Each computer serving as CAU groups 20 devices, so there are 5 distributed CAUs controlling 20 virtual devices each. Devices get their keys and signature and finally authenticate from their corresponding control area units (as described in Figure 5).

B. Comparison

In this section, we aim at comparing the key distribution and authentication processes, as defined in the DKMA workflow in both the centralized and distributed architectures. To that end and for the sake of comparison, we adapt the proposed DKMA workflow to deploy in a centralized architecture, as shown in Figure 7 and described next.

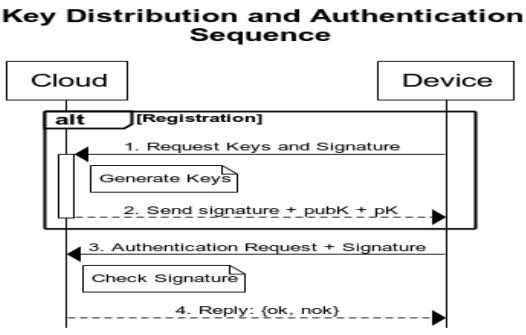

Fig. 7. Centralized key distribution and authentication workflow

Unlike the proposed distributed DKMA strategy the centralized approach consists in a unique step, as follows:

1. A fog device requests keys and signature to be authenticated.
2. The centralized key generation center at cloud generates keys and then sends public and private keys and signatures to the fog device through a secure channel.
3. Fog devices request authentication by sending their signatures
4. Finally, cloud verifies the signature and sends ACK to the devices.

The key distribution and signature process is done using ECDSA, as described in Section 2.

C. Results Analysis

We implement the two workflows in Python 3 and show the expected benefits for the proposed distributed approach in terms of key distribution and authentication delay, network delay, and network overhead, on the test-bed described above. The obtained results are described next.

Figure 8 shows the comparison of the two workflows for the key distribution and authentication delay. A substantial reduction in the delay for the proposed DKMA distributed approach is shown. Indeed, while the time grows exponentially with the number of devices for the centralized approach, it keeps almost flat for the distributed one, reaching the maximum reduction when considering 100 devices, from 28.69s to 1.0942s.

Similarly, Figure 9 shows the results obtained for the network delay, considering both workflows. The network delay is computed by dividing the Round Trip Time (RTT) by 2. We also show an incremental value for the delay reduction when using the distributed approach, reaching just the half (from 168ms to 84ms), when considering 100 devices.

Figure 10 shows a comparison of the network overhead for both approaches, in a table representing the whole set of messages in Kilo-Bytes (KB) forwarded throughout the network. We range the number of CAUs depending on the number of devices, assuming that, according to a certain policy in place, a CAU can manage up to 20 devices. Certainly, as shown in Figure 9, while the network overhead (KBs) keeps growing for the centralized approach as the number of devices increase, it will not change for the proposed DKMA distributed approach, as long as we can keep the assumed CAUs deployment policy.

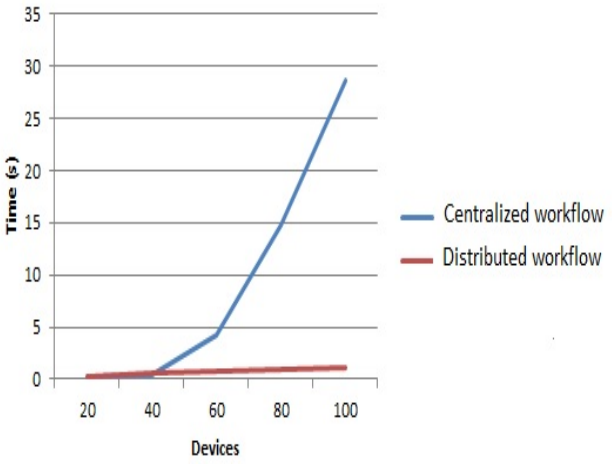

Fig. 8. Key distribution and authentication delay comparison.

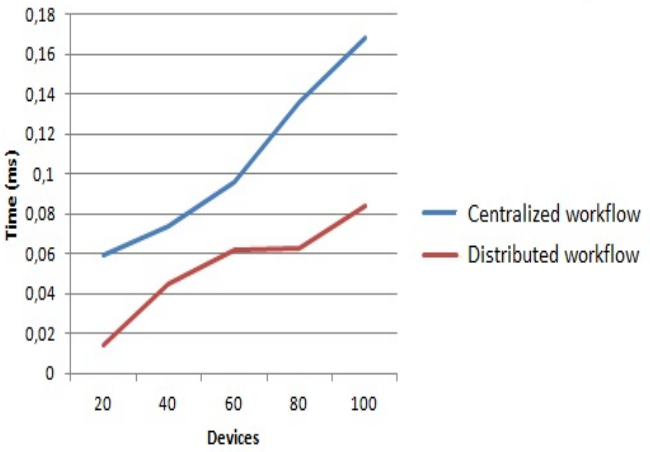

Fig. 9. Network delay comparison.

Devices	KB Centralized	CAU	KB DKMA
100	223,83	5	44,77
80	179,06	4	44,77
60	134,30	3	44,77
40	89,53	2	44,77
20	44,77	1	44,77

Fig. 10. Network overhead comparison (Kbytes).

Finally, we also analyze the total number of messages. To that end we only measure the number of messages, thus messages size is not considered. The total number of messages per device to get keys, signature and finally authenticate according both workflows (figure 4 and figure 5) and ECDSA algorithm in our implementation is 27 messages. In the first workflow, when using cloud as a centralized key distribution and authentication approach, the total number of messages goes and comes from cloud is (27*number of devices). However, when deploying the DKMA distributed controller's workflow, the number of messages is reduced up to (27*number of control area units), that is:

1. Centralized: 27*100=2700 number of messages
2. Distributed (DKMA): 27*5= 135 number of messages

As a summary of the obtained and presented results, we may assess that the proposed DKMA distributed approach for key distribution and authentication is much more efficient than a centralized one, while keeping the same security level.

VI. CONCLUSIONS

The Fog-to-cloud distributed hierarchical architecture imposes the need for a new coordinated security architecture and management to handle its distributed nature. In the literature we may find proposals using centralized key generator, centralized cloud key distribution, centralized trusted authority and other centralized tradition key distribution, that can even be enriched with replication and cloud elasticity to fix the single point of attack. However, despite using replication and cloud elasticity, the observed latency when moving to cloud, the added complexity and delay required by the key management process for thousands of devices, the huge amount of messages going to cloud or to the centralized key generator, and the whole network delay are all open challenges and issues, yet requiring further research efforts. This paper proposes a novel distributed key management and authentication strategy for F2C systems, referred to as the DKMA workflow, turning into significant benefits in terms of the time demanded by the key distribution and authentication processing, network delay, network overhead, and number of messages going to cloud. As a future work we plan to consider both hybrid key distribution and authentication to facilitate key management, as well as mobility (secure handover).

ACKNOWLEDGMENT

This work is supported by the H2020 CIPSEC (700378) and mf2c (730929) projects. For UPC authors it is also partially supported by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund under contract TEC2015-66220-R (MINECO/FEDER).

REFERENCES

[1] Jose A. Gonzalez-Martínez, Miguel L. Bote-Lorenzo, Eduardo Gomez-Sanchez, Rafael Cano-Parra, Cloud computing and education: A state-of-the-art survey, Computers & Education 80 (2015) 132-151

[2] Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, Tie Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, Journal of network and computer applications 2017

[3] X.Masip-Bruin, E.Marín-Tordera, A.Jukan, G.J.Ren, G.Tashakor, Foggy Clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud (F2C) computing systems, IEEE Wireless Communication Magazine,Vol. 23, issue 5, October 2016

[4] https://www.openfogconsortium.org/ [Accessed: January 2018]

[5] www.cipsec.eu [Accessed: January 2018]

[6] Anam Sajid , Haider Abbas and Kashif Saleem, Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges, special section on the plethora of research in internet of things (iot) IEEE ACCESS.2016.2549047

[7] Mikael Asplund, Simin Nadjm-Tehrani, Attitudes and Perceptions of IoT Security in Critical Societal Services, special section on the plethora of research in internet of things (iot), IEEE ACCESS.2016.2560919

[8] Sarang Kahvazadeh, Vitor B. Souza, Xavi Masip-Bruin, Eva Marín-Tordera, Jordi Garcia, Rodrigo Diaz, Securing combined Fog-to-Cloud system Through SDN Approach, CrossCloud'17, April 23, 2017, Belgrade, Serbia

[9] Jian Shen, Dengzhi Liu, Qi Liu, Bowei Wang, Zhangjie Fu, An Authorized Identity Authentication-based Data Access Control scheme in Cloud, ICACT 2016

[10] Hu Xiong, Kim-Kwang Raymond Choo, Athanasios V. Vasilakos, Revocable Identity-Based Access Control for Big Data with Verifiable Outsources Computing, IEEE Transactions on Big Data 2017

[11] Fagen Li, Bo Liu, Jiaojiao Hong, An efficient signcryption for data access control in cloud computing, Springer Computing (2017)

[12] I. Indu , P. M. Rubesh Anand , Shaicy P. Shaji, Secure File Sharing Mechanism and Key Management for Mobile Cloud Computing Environment, Indian Journal of Science and Technology 2016

[13] Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li, Albert Y. Zomaya, SeDaSC: Secure Data Sharing in Clouds, IEEE SYSTEMS JOURNAL, VOL. 11, NO. 2, JUNE 2017

[14] Marwa F. Mohamed, Service replication taxonomy in distributed environments, 4 January 2016 Springer-Verlag London 2016 DOI 10.1007/s11761-015-0189-7

[15] W.Delishiya Moral, B.Muthu Kumar, Improve the Data Retrieval Time and Security through Fragmentation and Replication in the Cloud, ICACCCT 2016

[16] V.Swathy, K.Sudha, R.Aruna, C.Sangeetha, R.Janani, Providing Advanced Security Mechanism for Scalable Data Sharing In Cloud Storage , ICICT 2016

[17] Renuka Prasad Pasupulati, Dr. Jordan Shropshire, Analysis of Centralized and Decentralized Cloud Architectures, SoutheastCon 2016

[18] Maged Hamada Ibrahim, Octopus: An Edge-Fog Mutual Authentication Scheme, International Journal of Network Security, Vol.18, No.6, PP.1089-1101, Nov. 2016

[19] Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu, Xiaoshuang Xing, Xiuzhen Cheng, An Attribute-Based Encryption Scheme to Secure Fog Communications, IEEE access 2017, DOI: 10.1109/ACCESS.2017.2705076

[20] Soumya Ranjan Moharana, Sourav Kanti Addya, Anurag Satpathy, Ashok Kumar Turuk, Banshidhar Majhi, V.K.Jha, Secure Key-distribution in IoT Cloud Networks, 2017 3rd International Conference on Sensing, Signal Processing and Security (ICSSS)

[21] X.Masip-Bruin, E.Marín-Tordera, A.Jukan, GJ.Ren, "Managing resources Continuity from the Edge to the Cloud: Architecture and Performance", Future Generation Computer Systems, in press, 10.1016/j.future.2017.09.036

[22] National Institute of Standards and Technology. 1999. Recommended Elliptic Curves for Federal Government, http://csrc.nist.gov/groups/ST/toolkit/documents/ dss/NISTReCur.pdf

[23] nvlpubs.nist.gov/nistpubs/FIPs

[24] Ruben Rios, Rodrigo Roman, Jose A. Onieva and Javier Lopez, From Smog to Fog: A Security Perspective, 2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)

[25] Bridget A. Martin, Frank Michaud, Don Banks, Arsalan Mosenia, Riaz Zolfonoon, Susanto Irwan, Sven Schrecker, John K. Zao, OpenFog Security Requirements and Approaches, Fog World Congress 2017